

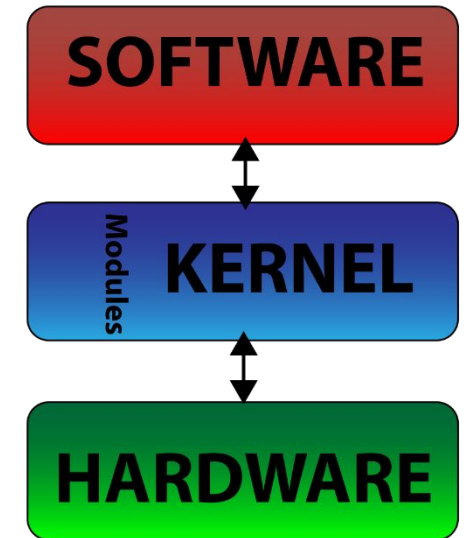
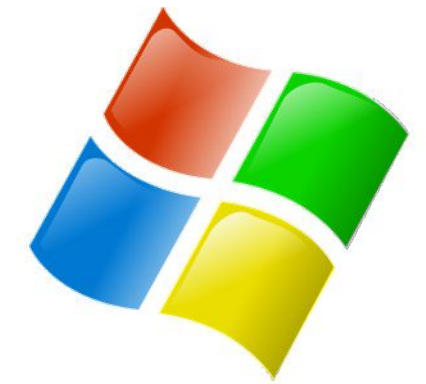


**UNIVERSITÀ DEGLI STUDI  
DELLA BASILICATA**

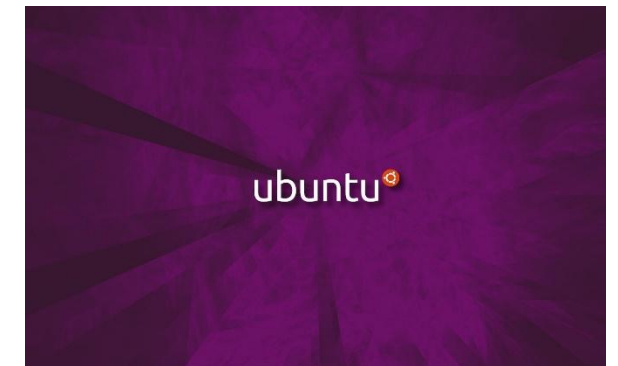
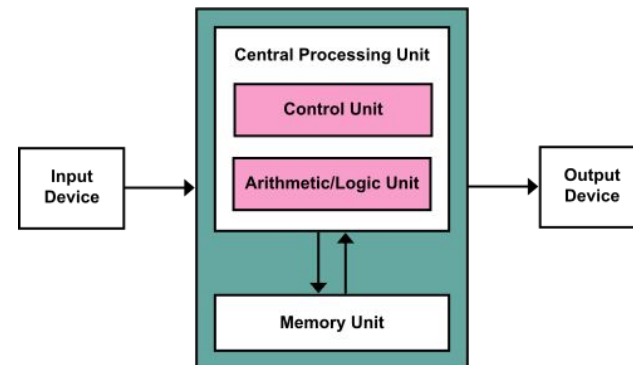
## *Corso di Sistemi Operativi*

# Esercitazione

## Gestione della memoria

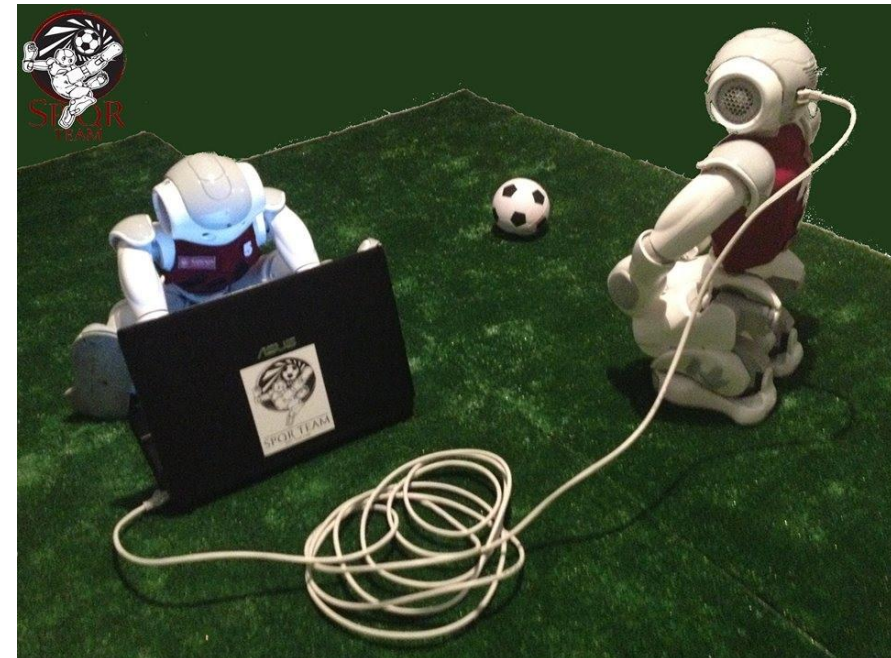
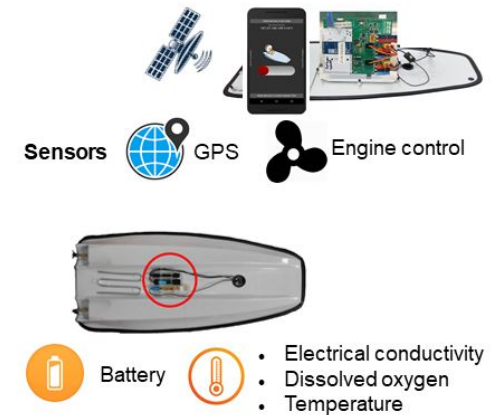


Docente:  
Domenico Daniele  
Bloisi



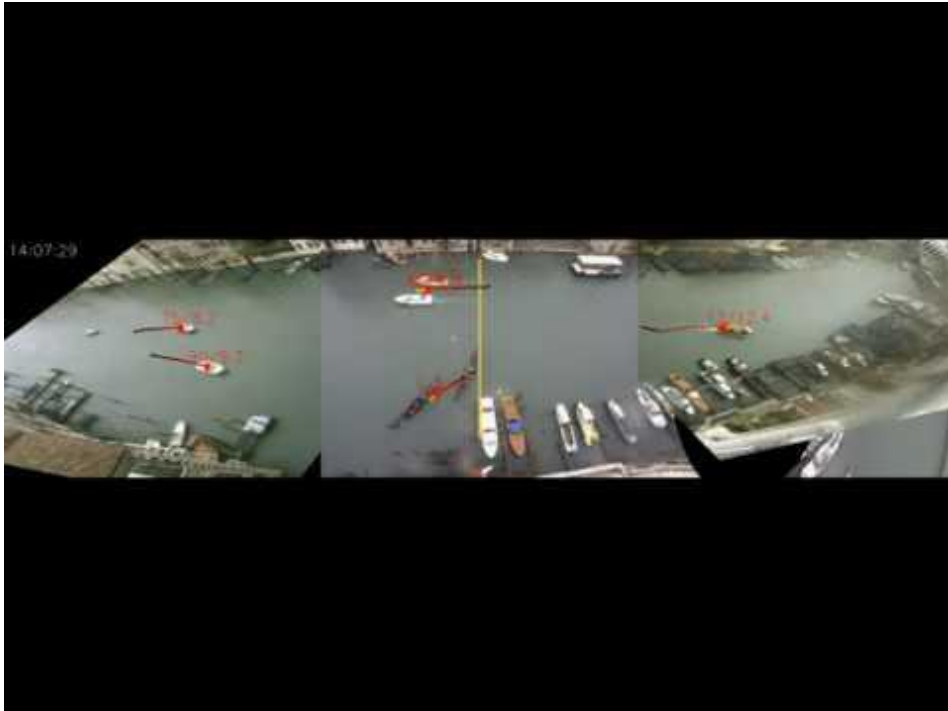
# Domenico Daniele Bloisi

- Professore Associato  
Dipartimento di Matematica, Informatica  
ed Economia  
Università degli studi della Basilicata  
<http://web.unibas.it/bloisi>
- SPQR Robot Soccer Team  
Dipartimento di Informatica, Automatica  
e Gestionale Università degli studi di  
Roma “La Sapienza”  
<http://spqr.diag.uniroma1.it>



# Interessi di ricerca

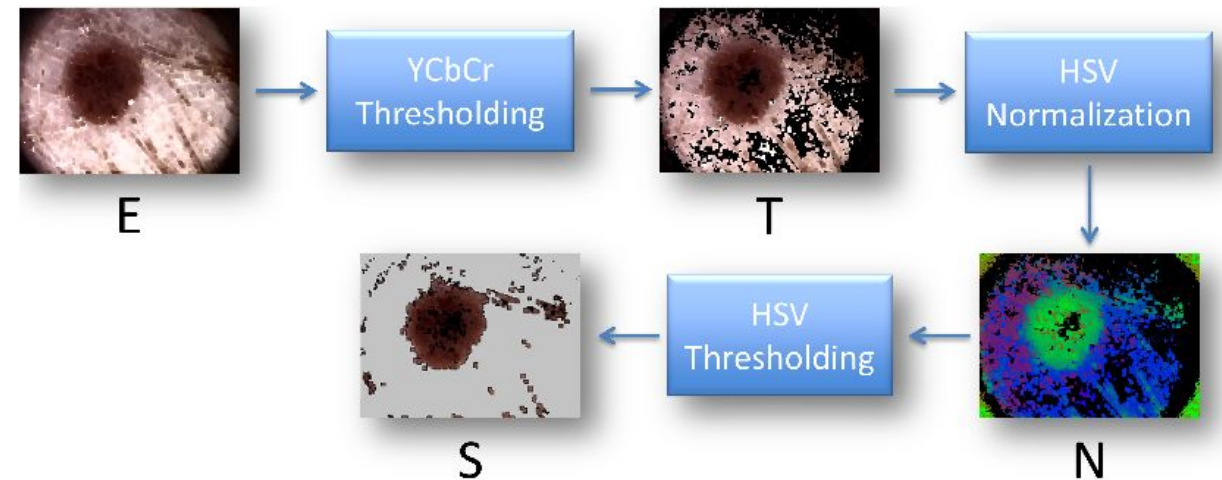
- Intelligent surveillance
- Robot vision
- Medical image analysis



[https://youtu.be/9a70Ucgbi\\_U](https://youtu.be/9a70Ucgbi_U)



<https://youtu.be/2KHNZX7UIWQ>



# UNIBAS Wolves <https://sites.google.com/unibas.it/wolves>



- UNIBAS WOLVES is the robot soccer team of the University of Basilicata. Established in 2019, it is focussed on developing software for NAO soccer robots participating in RoboCup competitions.

- UNIBAS WOLVES team is twinned with SPQR Team at Sapienza University of Rome



<https://youtu.be/ji0OmkaWh20>

# Informazioni sul corso

---

- Home page del corso:  
<http://web.unibas.it/bloisi/corsi/sistemi-operativi.html>
- Docente: Domenico Daniele Bloisi
- Periodo: I semestre ottobre 2022 – gennaio 2023
  - Lunedì dalle 15:00 alle 17:00 (Aula Leonardo)
  - Martedì dalle 08:30 alle 10:30 (Aula 1)

# Ricevimento

---

- In presenza, durante il periodo delle lezioni:  
Lunedì dalle 17:00 alle 18:00  
presso Edificio 3D, Il piano, stanza 15  
**Si invitano gli studenti a controllare regolarmente la bacheca degli avvisi per eventuali variazioni**
- Tramite google Meet e al di fuori del periodo delle lezioni:  
da concordare con il docente tramite email

Per prenotare un appuntamento inviare  
una email a  
[domenico.bloisi@unibas.it](mailto:domenico.bloisi@unibas.it)



# Credits

---

Alcuni esercizi derivano dai contenuti del corso

“Sistemi Operativi”

del Prof. Giorgio Grisetti

<https://sites.google.com/diag.uniroma1.it/sistemi-operativi-1819>

# Domanda 1

---

Con riferimento agli algoritmi di sostituzione delle pagine

1. enumerare i 4 principali algoritmi usati per tale scopo, ordinandoli in base al loro page-fault rate (dal più alto al più basso)
2. evidenziare gli algoritmi che soffrono dell'anomalia di Belady



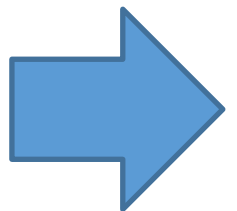
# Risposta Domanda 1

---

I quattro principali algoritmi di sostituzione delle pagine sono:

1. FIFO
2. Optimal page replacement (OPT)
3. Least recently used (LRU)
4. Second chance

L'algoritmo OPT garantisce il minimo tasso di page fault, seguito da LRU (che è una approssimazione di OPT). Second chance (FIFO modificato) produce un numero maggiore di page fault rispetto a LRU, ma inferiore rispetto a FIFO.



# Risposta Domanda 1

---

Gli algoritmi **FIFO** e **second chance** soffrono dell'**anomalia di Belady**, mentre OPT e LRU non sono soggetti all'anomalia di Belady.

Riassumendo:

ordine in base al page-fault rate	algoritmo	presenza dell'anomalia di Belady
1°	FIFO	SI
2°	second chance	SI
3°	LRU	NO
4°	OPT	NO

# Esercizio 1

---

Sia data la seguente successione di riferimenti alle pagine di memoria:

1, 2, 9, 2, 7, 4, 8, 9, 1, 2, 3, 9, 5, 1, 3

Si assuma

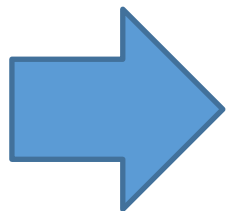
- di avere a disposizione **3 frame**, da gestire con politica optimal page replacement (**OPT**)
  - che  $T_{ma}$  e  $T_{pf}$  siano rispettivamente i tempi di accesso in memoria e di gestione di un page fault
1. Qual è il tempo di accesso effettivo in memoria per la situazione descritta?
  2. Qual è la probabilità di avere un page fault?

# Algoritmo di sostituzione OPT

---

**algoritmo ottimale di sostituzione delle pagine**

**(OPTimal page replacement):** *si sostituisce la pagina che non verrà usata per il periodo di tempo più lungo.*

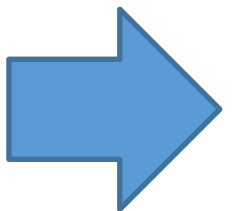


# Effective Access Time

---

*tempo di*  
*accesso*  
*effettivo ( $T_{EAT}$ )*

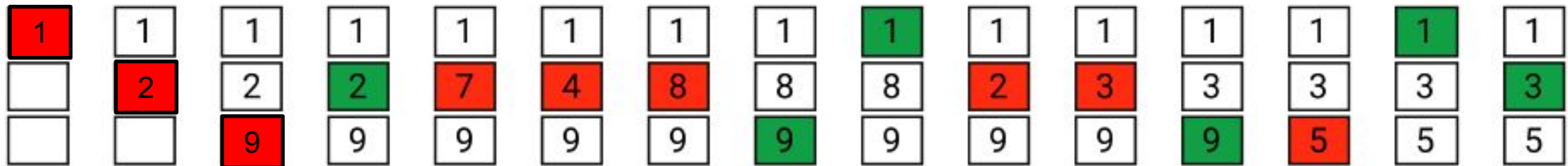
$$= \#\_page\_hit \times T_{ma} + \#\_page\_fault \times T_{pf}$$



# Soluzione Esercizio 1

---

L'evoluzione dello stato di occupazione dei frame per la successione di riferimenti è riportata sotto (in verde sono evidenziati i page hit, in rosso i page fault).



Il tempo effettivo di accesso (Effective Access Time) sarà

$$T_{\text{EAT}} = 6 \times T_{\text{ma}} + 9 \times T_{\text{pf}}$$

La probabilità di page fault sarà del 60%, ottenuta come

$$p_{\text{pf}} = 9 / 15 = 0.6$$

# Esercizio 2

---

Considerando la stessa situazione descritta nell'esercizio 1, calcolare di quanto aumentano le prestazioni incrementando il numero di frame a 4.

# Soluzione Esercizio 2

L'evoluzione dello stato di occupazione dei frame per la successione di riferimenti è riportata sotto (in verde sono evidenziati i page hit, in rosso i page fault).

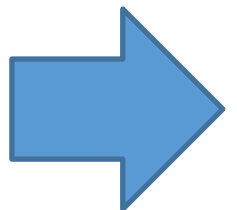
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2	2	2	2	5	5	5
		9	9	9	9	9	9	9	9	9	9	9	9	9
				7	4	8	8	8	8	3	3	3	3	3

Il tempo effettivo di accesso (Effective Access Time) sarà

$$T_{\text{EAT}} = 7 \times T_{\text{ma}} + 8 \times T_{\text{pf}}$$

La probabilità di page fault sarà del 53%, ottenuta come

$$p_{\text{pf}} = 8 / 15 = 0.53$$





# Soluzione Esercizio 2

---

Rispetto alla disponibilità di 3 frame, avendo a disposizione 4 frame la probabilità di page fault passa da 0.60 a 0.53, ottenendo, quindi, un miglioramento delle prestazioni pari al 7%

# Esercizio 3

---

Sia data la seguente successione di riferimenti alle pagine di memoria:

1, 3, 3, 2, 3, 7, 8, 3, 8, 1, 3, 8, 8, 3

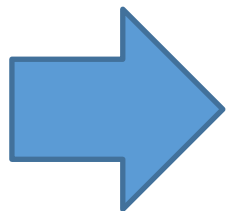
Si assuma

- di avere a disposizione **2 frame**, da gestire con politica Least Recently Used (**LRU**)
  - che  $T_{ma}$  e  $T_{pf}$  siano rispettivamente i tempi di accesso in memoria e di gestione di un page fault
1. Qual è il tempo di accesso effettivo in memoria per la situazione descritta?
  2. Qual è la probabilità di avere un page fault?

# Algoritmo di sostituzione LRU

---

La **sostituzione Least Recently Used (LRU)** associa a ogni pagina l'istante in cui è stata usata per l'ultima volta. Quando occorre sostituire una pagina, l'**algoritmo LRU** sceglie quella che non è stata usata per il periodo di tempo più lungo.



# Soluzione Esercizio 3

---

L'evoluzione dello stato di occupazione dei frame per la successione di riferimenti è riportata sotto (in verde sono evidenziati i page hit, in rosso i page fault).



Il tempo effettivo di accesso (Effective Access Time) sarà

$$T_{\text{EAT}} = 5 \times T_{\text{ma}} + 9 \times T_{\text{pf}}$$

La probabilità di page fault sarà del 64%, ottenuta come

$$p_{\text{pf}} = 9 / 14 = 0.64$$

# Esercizio 4

---

Considerando la stessa situazione descritta nell'esercizio 3, calcolare di quanto aumentano le prestazioni incrementando il numero di frame a 4.

# Soluzione Esercizio 4

L'evoluzione dello stato di occupazione dei frame per la successione di riferimenti è riportata sotto (in verde sono evidenziati i page hit, in rosso i page fault).

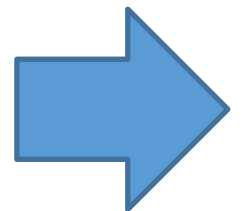
1	1	1	1	1	1	8	8	8	8	8	8	8	8
	3	3	3	3	3	3	3	3	3	3	3	3	3
			2	2	2	2	2	2	1	1	1	1	1
					7	7	7	7	7	7	7	7	7

Il tempo effettivo di accesso (Effective Access Time) sarà

$$T_{\text{EAT}} = 8 \times T_{\text{ma}} + 6 \times T_{\text{pf}}$$

La probabilità di page fault sarà del 42%, ottenuta come

$$p_{\text{pf}} = 6 / 14 = 0.42$$



# Soluzione Esercizio 4

---

Rispetto alla disponibilità di 2 frame, avendo a disposizione 4 frame la probabilità di page fault passa da 0.64 a 0.42, ottenendo, quindi, un miglioramento delle prestazioni pari al 22%

# Esercizio 5

---

Sia data la seguente successione di riferimenti alle pagine di memoria:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 1, 3, 7, 6, 3

Si assuma

- di avere a disposizione **3 frame**, da gestire con politica **FIFO**
  - che  $T_{ma}$  e  $T_{pf}$  siano rispettivamente i tempi di accesso in memoria e di gestione del page fault
1. Qual è il tempo di accesso effettivo in memoria per la situazione descritta?
  2. Qual è la probabilità di avere un page fault?



# Esercizio 6

---

Sia data la seguente successione di riferimenti alle pagine di memoria:

7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4

Si assuma

- di avere a disposizione **4 frame**, da gestire con politica **FIFO**
  - che  $T_{ma}$  e  $T_{pf}$  siano rispettivamente i tempi di accesso in memoria e di gestione del page fault
1. Qual è il tempo di accesso effettivo in memoria per la situazione descritta?
  2. Qual è la probabilità di avere un page fault?

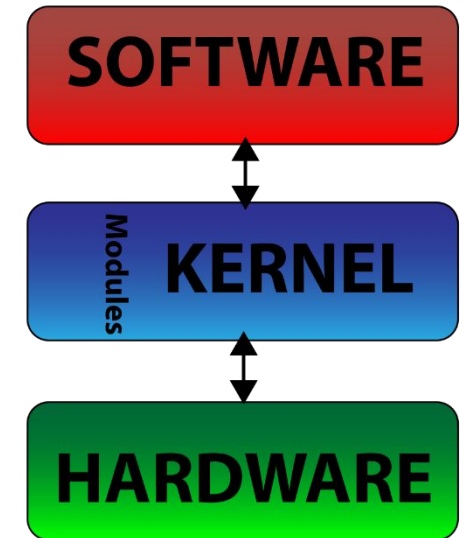
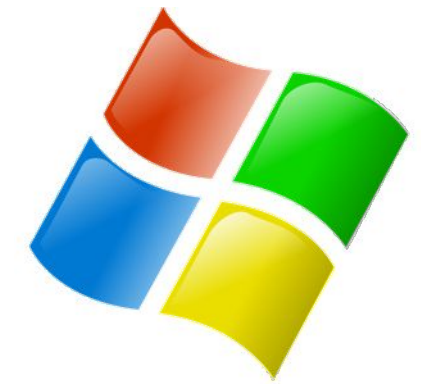


**UNIVERSITÀ DEGLI STUDI  
DELLA BASILICATA**

*Corso di Sistemi Operativi*

# Esercitazione

## Gestione della memoria



Docente:  
Domenico Daniele  
Bloisi

